

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of the Claims:

1. **(Currently Amended)** A computer-implemented method for increasing the performance of a virtual machine, the computer-implemented method comprising:
 - obtaining a program instruction to be executed by the virtual machine;
 - determining ~~when~~ **whether** the program instruction is a branch instruction;
 - determining ~~when~~ **whether** a basic block is present in a code cache when it is determined that the program instruction is a branch instruction, the basic block including code **that represents the program instruction**, the code cache being associated with the virtual machine; and
 - executing the code included in the basic block when it is determined that the basic block is present **in the code cache and the program instruction is a branch instruction**.
2. **(Currently Amended)** A computer-implemented method as recited in claim 1, wherein when it is determined that the basic block is not present in the code cache, the method further includes:
 - interpreting the program instruction; and
 - copying code corresponding to the program instruction into the code cache.
3. **(Original)** A computer-implemented method as recited in claim 2 further including:
 - allocating space in the code cache for the code corresponding to the program instruction; and
 - providing the code corresponding to the program instruction with a label.
4. **(Original)** A computer-implemented method as recited in claim 3 further including placing the label in a table of labels.

5. **(Currently Amended)** A computer-implemented method as recited in claim 2 wherein determining ~~when~~ whether the basic block is present in the code cache includes searching through a table of labels to determine if a target associated with the program instruction has a ~~substantially~~ matching label in the table of labels.

6. (Original) A computer-implemented method as recited in claim 2 wherein the program instruction is a bytecode, and wherein the bytecode is executed by an interpreter of the virtual machine.

7. (Original) A computer-implemented method as recited in claim 2 wherein the code cache is a native code cache, and the code corresponding to the program instruction is native code.

8. (Original) A computer-implemented method as recited in claim 1 wherein the program instruction is a bytecode and the code cache is a native code cache.

9. **(Currently Amended)** A computer-implemented method as recited in claim 8 further including interpreting the bytecode when the determining determines ~~it is determined~~ that the program instruction is not ~~the~~ a branch instruction.

10. **(Currently Amended)** A computer-implemented method as recited in claim 1 further including:

computing a target using the program instruction, wherein the determining of whether ~~when~~ the basic block is present in the code cache includes determining if the code cache includes any basic blocks which correspond to the target.

11. **(Currently Amended)** A computer program product for increasing the performance of a virtual machine, the computer program product comprising:

computer code for obtaining a program instruction to be executed by the virtual machine;

computer code for determining ~~when~~ whether the program instruction is a branch instruction;

computer code for determining ~~when~~ whether a basic block is present in a code cache when it is determined that the program instruction is a branch instruction, the basic block including code, the code cache being associated with the virtual machine;

computer code for executing the code included in the basic block when it is determined that the basic block is present and the program instruction is a branch instruction; and

a computer-readable medium that stores the computer codes.

12. (Currently Amended) A computer program product as recited in claim 11, wherein when it is determined that the basic block is not present in the code cache, the computer program product further includes: ~~further including:~~

computer code for interpreting the program instruction when it is determined that the basic block is not present in the code cache; and

computer code for copying code corresponding to the program instruction into the code cache when it is determined that the basic block is not present in the code cache.

13. (Original) A computer program product as recited in claim 12 further including:

computer code for allocating space in the code cache for the code corresponding to the program instruction;

computer code for providing the code corresponding to the program instruction with a label; and

computer code for placing the label in a table of labels.

14. (Original) A computer program product as recited in claim 11 wherein the program instruction is a bytecode and the code cache is a native code cache.

15. (Original) A computer program product as recited in claim 11 wherein the computer-readable medium is one selected from the group consisting of a data signal embodied in a carrier wave, a floppy disk, a computer memory, a hard disk, an optical disk, a tape drive, and a CD-ROM.

16. (Currently Amended) A computing system, the computing system including a virtual machine, the virtual machine comprising:

a code cache; and

an interpreter, the interpreter being arranged to obtain[[ing]] a bytecode, the interpreter further being arranged to determining ~~when~~ whether the bytecode is a branch bytecode and to determine when a basic block is present in the code cache when it is determined that the bytecode is a branch bytecode, the basic block including native code, wherein the interpreter causes the native code to be executed when it is determined that the basic block is present and the program instruction is a branch instruction.

17. (Currently Amended) A computing system including a virtual machine according to claim 16 wherein the interpreter is further arranged to :

interpret the bytecode when the determining determines ~~it is determined~~ that the basic block is not present in the code cache ;and

to copy native code corresponding to the bytecode into the code cache when the determining determines that the basic block is not present in the code cache.

18. (Currently Amended) A computing system including a virtual machine according to claim 16 wherein the interpreter is further arranged to interpret the bytecode when the determining determines ~~it is determined~~ that the bytecode is not a branch bytecode.

19-24. (Canceled)

25. (New) A computer-implemented method as recited in claim 1, wherein when the determining determines that the program instruction is not a branch instruction, the method further comprises:

interpreting the program instruction; and

copying code corresponding to the program instruction into the code cache.

26. (New) A method for increasing the performance of a virtual machine, the method comprising:

reading a program instruction to be executed by the virtual machine;

(a) determining whether the program instruction is a branch instruction;

(b) determining whether a basic block is present in a code cache when the determining (a) determines that the program instruction is a branch instruction, wherein the basic block including code that represents the program instruction;

executing the code included in the basic block when the determining (a) determines that the basic block is present in the code cache and the determining determines that program instruction is a branch instruction;

interpreting the program instruction when the determining (a) determines that the program instruction is not a branch instruction; and

copying code corresponding to the program instruction into the code cache when the determining (a) determines that the program instruction is not a branch instruction.

27. (New) A virtual machine capable of:

reading a program instruction to be executed by the virtual machine;

(a) determining whether the program instruction is a branch instruction;

(b) determining whether a basic block is present in a code cache when the determining (a) determines that the program instruction is a branch instruction, wherein the basic block including code that represents the program instruction;

executing the code included in the basic block when the determining (a) determines that the basic block is present in the code cache and the determining determines that program instruction is a branch instruction;

interpreting the program instruction when the determining (a) determines that the program instruction is not a branch instruction; and

copying code corresponding to the program instruction into the code cache when the determining (a) determines that the program instruction is not a branch instruction.